順序ソート型付ラムダ計算における簡約と単一化 (アルゴリズムと計算の理論)

原尾, 政輝

数理解析研究所講究録
1041巻 1998年 219-226

順序ソート型付ラムダ計算における簡約と単一化

九州工業大学　情報工学部　原尾　政輝 (Masateru HARAO)[1*]

あらまし　単純型付ラムダ計算の型に順序関係を導入して拡張した順序ソート型付ラムダ計算について，階層的知識表現や順序ソート高階単一化による推論処理の定式化の観点から考察する。まず，制限関数の概念に基づいた部分型関係と代入規則を導入し，変数変換によって継承機構を実現するラムダ計体系Λ*を定義する。この拡張された体系では通常の反変的な部分型関係の下でのラムダ簡約における非整合性や項の等価性の記述の問題などを解消でき，自然なスキーマ表現と高階単一化が定式化可能であることを示す。

1 順序ソート型理論

1．1項，型および部分型

基本ソートの集合をS_0とするとき，単純型Sとは$S_0 \subseteq S$であって$\sigma, \tau \in S$ならば$\sigma \rightarrow \tau \in S$となる$S_0$を含む最小の集合であるとする。各型$\sigma \in S$に対して定数および変数の可算集合$C_\sigma$,$V_\sigma$が存在し，項tが型$\tau$を持つ事を$t:\tau$または$t_\tau$で表す。$S_0$間には半順序関係$\leqq \subseteq S_0 \times S_0$が定義されており，$S$に拡張された順序関係$\leqq$を**部分型関係**，順序関係$\leqq$を導入した型を**順序ソート型**あるいは単に**順序型**と呼ぶ。項への型付けはChurch流の明示的方式[2]に従い，次の単純型システムの型推論規則および新しく導入する型変換規則を用いて行う。

定義1　型推論規則

(1)**導入**　[定数] $\dfrac{c \in C_\sigma}{c_\sigma}$　[変数] $\dfrac{x \in V_\sigma}{x_\sigma}$　(2)**抽象化** $\dfrac{x_\tau \qquad t_\sigma}{\lambda x_\tau.t : \tau \rightarrow \sigma}$

(3)**適用** $\dfrac{f_{\tau \rightarrow \sigma} \qquad t_\tau}{f_{\tau \rightarrow \sigma} t_\tau : \sigma}$

部分型関係としては形式的に次のようなものが考えられる。

（1）反変的順序関係　：$\sigma \rightarrow \tau <: \sigma' \rightarrow \tau' \Leftrightarrow \sigma' \leqq \sigma$　かつ$\tau <: \tau'$
（2）正規的順序関係　：$\sigma \rightarrow \tau \leqq: \sigma' \rightarrow \tau' \Leftrightarrow \sigma' = \sigma$　かつ$\tau \leqq: \tau'$
（3）共変的順序関係　：$\sigma \rightarrow \tau \ll: \sigma' \rightarrow \tau' \Leftrightarrow \sigma \leqq \sigma'$　かつ$\tau \ll: \tau'$

以下一般的な部分型を$\leqq$で，反変，正規，共変を$<:$，$\leqq:$，$\ll:$でそれぞれ表わす。部分

[1*]〒820850 飯塚市川津680-4，e-mail:harao@dumbo.ai.kyutech.ac.jp

型関係の解釈としては種々のものが可能であるが[3,4,15]，データや知識の表現としての立場から集合包含関係として解釈する。各型τには**意味**[τ]が定義されており，基本型$\sigma \in \boldsymbol{S}_0$に対する意味[$\sigma$]は各解釈に共通とし【$\sigma$】で表わし，基本となるデータの集合を意味するとする。関数型$\sigma \to \tau$の意味[$\sigma \to \tau$]は[σ]から[τ]へのある関数の集合とする。部分型関係≦はこの集合の包含関係として次のように解釈する。

条件　部分型関係と型の解釈。

（１）$\sigma, \sigma' \in \boldsymbol{S}_0$に対して[$\sigma$]，[$\sigma'$]は集合で，$\sigma \leqq \sigma'$ならば[$\sigma$]⊆[$\sigma'$]となる.

（２）$\sigma \to \tau \leqq \sigma' \to \tau' \Leftrightarrow [\sigma \to \tau] \subseteq [\sigma' \to \tau']$

１．２　型継承規則と簡約

部分型における型継承規則は，一般に次のように定義される[4]。

(A)型変換規則(Coerce)　$\dfrac{f_\sigma \quad \sigma \leqq \sigma'}{f_{\sigma'}}$

この規則による型付けは集合への所属関係として解釈することができる。すなわち，型付き項t_σは$t \in [\sigma]$と見なし，型変換規則はt_σ（$t \in [\sigma]$）のとき，$\sigma \leqq \tau$なら$t \in [\sigma] \subseteq [\tau]$だから$t_\tau$と解釈する。例えば，nat≦intのとき，c∈[nat]⊆[int]よりc_{nat}ならばc_{int}となる。

型変換の下での自由変数x_σへの**代入**θとは関数$\boldsymbol{V} \to \boldsymbol{L}$であって，有限の$x \in \boldsymbol{V}$に対して，$\theta(x) \neq x$を満たす，$x_\sigma$なら$\theta(x_\sigma)$:$\sigma$，となる型を保存する写像である。以下，自由変数$x_\sigma$への$t_\sigma$の代入を$[x_\sigma := t_\sigma]$などとも書く。次の$\alpha$，$\beta$および$\eta$-簡約と型継承規則をもつ計算体系を**順序ソート単純型付ラムダ計算**と呼ぶ。

定義２（α，βおよびη簡約）

（１）α-簡約: $\lambda x_\tau.s \Rightarrow_\alpha \lambda y_\tau.s[x_\tau := y]$，$y_\tau \notin FV(s)$

（２）β-簡約:$(\lambda x_\tau.s)t_\tau \Rightarrow_\beta s[x := t_\tau]$.

（３）η-簡約: $\lambda x_\tau.sx_\tau \Rightarrow_\eta s$,

型変換規則を導入した体系では同じ項が異なる型付けを持つことになり，いわゆる**多義性**(overloaded)をもつ。関数$f:\sigma \to \tau$の**定義域**，**値域**をそれぞれdef(f)，ran(f)で表わすとき,. 多義性をもつ関数ではdef(f)＝[σ]となるとは限らなず，型の情報は関数の性質を厳密に規定できなくなってしまう。そのため，[σ]＝def(f),[τ]＝ran(f)となる型付けを特に**標準的型付け**と呼び$t::\sigma \to \tau$で表わす。

項を$f::\sigma \to \tau$，関数変数を$F_{\sigma' \to \tau'}$, $\sigma \to \tau \leqq \sigma' \to \tau'$,とする。fは$f_{\sigma' \to \tau'}$と型変換によっ

て型付け可能だから$F_{\sigma'\rightarrow\tau'}$に代入できる。項$F_{\sigma'\rightarrow\tau'}(c_{\sigma'})$へのfの代入によって$f(c_{\sigma'})$を得る。この適用が定義可能なためには$\sigma' \leqq \sigma$である必要がある。すなわち，$\sigma\rightarrow\tau \leqq \sigma'\rightarrow\tau'$ならσ'≦σだから≦は反変的部分型関係<:でなければならない。このように型変換規則(A)による継承では反変的部分型になり，一般に次のようにの解釈される[4,15]。

$$《\sigma\rightarrow\tau》=\{f \mid \mathrm{def}(f)\supseteq《\sigma》, \forall x\in《\sigma》\text{に対して}g(x)\in《\tau》\text{なる関数}\}$$

すなわち，$《\sigma\rightarrow\tau》$は少なくとも$《\sigma》$の要素に対して定義された関数の集合を表す。

1.3 順序ソート型付きラムダ計算と問題点

$f::\sigma\rightarrow\tau$とするとき，そのη正規形に対する反変的部分型関係の下での次のような型変換を考える。

$$\frac{\lambda x_\sigma.f(x)::\sigma\rightarrow\tau \quad \sigma\rightarrow\tau <: \sigma'\rightarrow\tau}{(\lambda x_\sigma.f(x)):\sigma'\rightarrow\tau}$$

得られた型付けでは，λ抽象したx_σの型と型付け$\sigma'\rightarrow\tau$の領域型とが一致しない。これは，型付けが集合所属関係$(\lambda x_\sigma.f(x))\in[\sigma'\rightarrow\tau]$の意味で，$\lambda x_{\sigma'}$と領域を$\sigma'$に制限する意味ではないことに起因する。しかし，$(\lambda x_{\sigma'}.f(x)):\sigma'\rightarrow\tau$と見なした方が自然である。

型変換規則は，抽象化変数に対しては制限的に，本体の変数には拡張的に型変換するので不自然な型付けを引き起こす。例えば，恒等写像$t=\lambda x_\sigma.x:\sigma\rightarrow\sigma$に対して，$\sigma' <: \sigma$のとき$\sigma\rightarrow\sigma <: \sigma'\rightarrow\sigma$だから，型変換によって$\lambda x_\sigma.x:\sigma'\rightarrow\sigma$，が得られるが，恒等写像の型付けとしては不自然である。しかし，$\sigma\rightarrow\sigma <: \sigma'\rightarrow\sigma'$なる順序はつかないから$\lambda x_\sigma.x:\sigma'\rightarrow\sigma'$ともできない。また，変数に対する型変換規則の適用は，変数の意味する領域を変化させてしまうから簡約においても非整合性を誘引する。

ラムダ項$\lambda x_\sigma.(\lambda y_\tau.y)x_\sigma$の簡約を考えよう。$\eta$-簡約規則，$\beta$-簡約規則をそのまま用いると，

$$\lambda y_\tau.y:\tau\rightarrow\tau \;{}_\eta\!\Leftarrow \lambda x_\sigma.(\lambda y_\tau.y)x_\sigma \Rightarrow_\beta \lambda x_\sigma.x_\tau:\sigma\rightarrow\tau,\quad \sigma <: \tau$$

となり[2*]このままでは合流性を満たさないし型付けも非整合的である。この問題を解消するためにCurien等[8]は,多相型において項書き替え系の手法を用いて考察している。しかしながら，この手法では後述する項の同等性や単一化などに対しては有効でない。

また，多義性を持つ関数では次の，$f:\sigma\rightarrow\tau, f:\sigma'\rightarrow\tau'$と型付け可能なとき，（1）fは$[\sigma]\cap[\sigma']$では一致する，（2）単調性：$\sigma \leqq \sigma'$なら$\tau \leqq \tau'$，は自然な仮定である[13]。しかし，反変的な型では，この単調性は成り立たない。

2* ここでは$(\lambda y_\tau.y)x_\sigma \rightarrow^{\mathrm{coerce}} (\lambda y_\sigma.y)_{\sigma\rightarrow\tau}x_\sigma \rightarrow^\beta y[y:=x_\sigma]=x_\tau$と考える。代入結果の型付けを，$y[y:=x_\sigma]=x_\sigma$としても非整合性が生じることに注意する。

反変的部分型の下での継承では，関数fとgを$f_{\sigma\to\tau}$かつ$g_{\sigma\to\tau}$であって$\forall x\in[\sigma]$に対して$f(x)=g(x)$であるとしても$f=g$とは限らず，項の同等性を規定できない[15]。例えば，恒等写像を$id_{int\to int}$，絶対値関数を$abs_{int\to int}$とする。継承によって$id_{nat\to int}$，$abs_{nat\to int}$なる型付けが可能であり，$\forall x\in[nat], id(x)=abs(x)$が成り立つ。しかし，正規な型付けの下での関数としては $id::int\to int\neq abs::int\to int$が成り立つので，$id_{nat\to int}\neq abs_{nat\to int}$である。

2 制限関数に基づく継承の定式化

2．1　制限関数と制限変数変換規則

項fの定義域を$\sigma'\leqq\sigma$なるσ'に制限して得られる関数を$f|_{\sigma'}=\lambda y_{\sigma'}.f_{\sigma\to\tau}(y_{\sigma'})$で表わし，丁度定義域$\sigma'$を持つ項と解釈する。また，$f_{\sigma\to\tau}$と$f|_{\sigma'}$は全く異なる定義域を持った異なる関数であると見なす。そして，$g::\sigma'\to\tau'$のときあるfに対して，$\forall x\in[\sigma'],f(x)=g(x)$なる関係が成り立つときgはfの制限関数という。fの制限関数$f|_{\sigma'}$は正確には$g_{\sigma'\to\tau'}=\lambda x_{\sigma'}.f(x_{\sigma'})$とfとgは異なる関数と見るべきであるが，混乱が生じないときはこれを，$f|_{\sigma'}=f_{\sigma'\to\tau'}=\lambda y_{\sigma'}.f_{\sigma\to\tau}(y)$とも書くことにする。また，制限関数のラムダ表現では，本体の変数も$y_{\sigma'}$へ変換されていることに注意する。

反変的部分型による継承の最大の問題点は，型付けを集合への所属として解釈することにある。そのため，関数の型変換は制限関数として解釈する。例えば，

$$\frac{\lambda x_{\sigma}.f(x)::\sigma\to\tau \quad \sigma'\leqq\sigma}{(\lambda x_{\sigma'}.f(x)):\sigma'\to\tau}$$

は丁度定義域σ'を持つ関数$\lambda x_{\sigma'}.f(x):\sigma'\to\tau'$と見なし，型付けは正規的であるとする。しかし正規的型付けでは，$\sigma\neq\sigma'$ならば$[\sigma\to\tau]\cap[\sigma'\to\tau']=\phi$となり型変換規則が適用できない。そのため，制限変数変換の概念を新しく導入する。

変数x_{σ}の$y_{\sigma'}$，$\sigma'\leqq\sigma$，への変換を$x_{\sigma}\Rightarrow y_{\sigma'}$と書き，**制限変数変換**と呼ぶ。関数変数の制限関数についても，$F_{\sigma\to\tau}=\lambda x_{\sigma}.F(x)\Rightarrow\lambda y_{\sigma'}.F(y):\sigma'\to\tau'=F_{\sigma'\to\tau'}$と考え，これを$F_{\sigma\to\tau}\Rightarrow F_{\sigma'\to\tau'}$と書く。このとき，変数変換$x_{\sigma}\Rightarrow y_{\sigma'}$は$\sigma'\leqq\sigma$のとき可能だから，$\sigma'\to\tau'\leqq\sigma\to\tau$ならば$\sigma'\leqq\sigma$，$\tau'\leqq\tau$となり共変的部分型となる。また，正規的型付けの関数集合を【$\sigma\to\tau$】で表すとき，この型の意味は次のように与えることができる。

$$《\sigma\to\tau》=\{f\mid g|_{\sigma'}=f,\ \exists g\in【\sigma\to\tau】,\sigma'\ll\sigma\}$$

関数変数の変数変換の可能性は部分式に含まれる変数の型にも依存する。式tにおける変数Fをヘッドとする部分項，もしがそれが基本型でなければ適当な変数を代入して基本型まで簡約したもの，の成分を変数Fの**文脈**という。例えば，$t(\ldots,F_{\sigma\to\tau\to\gamma}(c_{\sigma}),\ldots):\gamma$において$(c_{\sigma},x_{\tau})$は$F_{\sigma\to\tau\to\gamma}$の文脈である。変数$F_{\sigma}$の文脈を$\Gamma$，文脈$\Gamma$に含まれる各自

由変数$x_{\tau 1},\ldots,x_{\tau k}$を制限変数変換して得られるFの型を$\sigma'$とするとき，$F_{\sigma'}$を関数変数$F_\sigma$の文脈$\Gamma$の下での$\sigma'$への制限変数変換という．$F_\sigma$と$F_{\sigma'}$は異なる関数であるが，混乱が生じない限りこのように書く．例えば，

$$\lambda x_{\tau'}.F_{\sigma\to\tau\to\gamma}(c_\sigma,x_{\tau'}),\ \tau'\ll:\tau$$

はx_τのτ'への制限変数変換で，$F_{\sigma\to\tau\to\gamma}\Rightarrow H_{\sigma\to\tau'\to\gamma}$は文脈$(c_\sigma, x'_\tau)$の下での制限関数変換である。

2.2 制限変換規則と簡約

変数x_σへのt_τ，$\sigma\neq\tau$，の代入は，制限変数変換によって$x_\sigma\Rightarrow y_\tau$と，変数の型を項の型へ変換できるときとし，$(x_\sigma[x_\sigma\Rightarrow y_\tau])[y_{\sigma'}:=t_\tau]$を$\theta=[x_\sigma:=^* t_\tau]$で表す。

定義　順序代入θとは，関数$V\to L$であって，有限の$x\in V$に対して$\theta(x)\neq x$であって，$x_\sigma\Rightarrow y_\tau$，$\tau\ll:\sigma$，なら$\theta(x_\sigma):\sigma'$，$\theta=[x_\sigma:=^* c_{\sigma'}]$，$\sigma'\ll:\sigma$，となるものである。

制限変数変換規則に基づく項の制限変換規則を次のように与える。

(B)項の制限変換規則　$$\frac{\lambda x_\sigma.f:\sigma\to\gamma \qquad \tau\ll:\sigma}{\lambda y_\tau.f:\tau\to\gamma'}(\Rightarrow^*)$$

ただし，γ'は$x_\sigma\Rightarrow^* y_\tau$により，$\gamma$中の変数$x_\sigma$の型を$\tau$に変換して得られる型で$\gamma'=\gamma[\sigma\Rightarrow^*\tau]$で表わす。

例　項の制限変換

（1）$\lambda x_\sigma.f\,x:\sigma\to\gamma\Rightarrow^*\lambda y_\tau.f\,x[x:=^* y_\tau]:\sigma\to\gamma=\lambda y_\tau.f\,y:\sigma\to\gamma'$

（2）$\lambda x_\sigma.(\lambda F_{\sigma\to\tau\to\gamma}.F(c_\sigma,x_\tau))\Rightarrow^*\lambda y_{\tau'}.(\lambda F_{\sigma\to\tau\to\gamma}.F(c_\sigma,x_\tau[x:=^* y_{\tau'}]))$

$$\Rightarrow^*\lambda y_{\tau'}.(\lambda F_{\sigma\to\tau'\to\gamma}.F(c_\sigma,y_{\tau'})),\ \tau'\ll\tau$$

順序代入を用いた，α^*簡約，β^*簡約，η^*簡約を次のように定義し，その下でのラムダ計算体系をΛ^*で表す。

定義　Λ^*体系における簡約規則

α^*-簡約　$\lambda x_\sigma.t:\gamma\Rightarrow_{\alpha^*}\lambda y_{\sigma'}.t[x:=^* y_{\sigma'}]:\gamma[\sigma\Rightarrow^*\sigma'],\ x_\sigma\Rightarrow y_{\sigma'}$。

β^*-簡約　$(\lambda x_\sigma.t)c_{\sigma'}:\gamma\quad\Rightarrow_{\beta^*}\ (t[x_{\sigma'}:=^* c_{\sigma'}]):\gamma[\sigma\Rightarrow\sigma']\quad,\sigma'\ll\sigma$

η^*-簡約: $\lambda x_\sigma.t(x)\Rightarrow_{\eta^*}t\quad\Leftrightarrow\lambda x_\sigma.t(x):\sigma\to\tau$かつ$t:\sigma\to\tau$

ただし，α^*-簡約でのγ'は$x_\sigma \Rightarrow^* y_{\sigma'}$により，$\gamma$中の変数の型を$\sigma'$に変換して得られる型を表わす。$\eta^*$簡約では，tと$\lambda x_\sigma.t(x)$とが同じ型付けのときのみ簡約を許す。

2.3　Λ^*における計算の性質

体系をΛ^*では，型変換(Coerce)規則は許さず，継承は制限変数変換のみによって実現するとする。順序代入の下では，x_σには$\tau \ll: \sigma$とするとき$x_\sigma \Rightarrow x_\tau$なる任意の$c_\tau, y_\tau$が代入可能である。これは$\beta$簡約規則とも整合的であるし，関数と型付けの遊離も解消出来る。例えば，fとして$\lambda x_\tau.x:\tau \to \tau$をとると，$\lambda x_\tau.x:\tau \to \tau \Rightarrow \lambda y_\sigma.x[x:=^*y] = \lambda y_\sigma.y$ だから，

$$\dfrac{\dfrac{\lambda x_\tau.x:\tau \to \tau \qquad \sigma \ll:: \tau}{(\lambda y_\sigma.y):\sigma \to \sigma}(\Rightarrow) \qquad z_\sigma}{z_\sigma}$$

とすることが可能である。また，共変的な型付けであるから単調性も満たす。

ラムダ項$\lambda x_\sigma.(\lambda y_\tau.y)x_\sigma$の簡約を考えよう。$\beta^*$-簡約規則を用いると，

$$(\lambda y_\tau.y)x_\sigma \Rightarrow^* (\lambda y_\sigma.y)x_\sigma \to_\beta y[y:=x_\sigma] = x_\sigma$$

だから，$\lambda y_\sigma.y:\sigma \to \sigma \;{}_{\eta^*}\!\Leftarrow \lambda x_\sigma.(\lambda y_\tau.y)x_\sigma \Rightarrow_{\beta^*} \lambda x_\sigma.x:\sigma \to \sigma$，$\sigma \ll: \tau$となる。

命題1　Λ^*はβ^*-簡約，η^*-簡約の下で合流性を満たす.

制限関数では項の同等関係を正確に記述できる。このとき，$f:\sigma \to \tau \neq g:\sigma \to \tau$であっても，$\sigma' \ll: \sigma$なる$\sigma'$に対して$\sigma'$への制限関数関数はまったく異なる関数と見なすから$\lambda x_{\sigma'}.f(x_{\sigma'}):\sigma' \to \tau = \lambda x_{\sigma'}.g(x_{\sigma'}):\sigma' \to \tau$のような等式関係が成り立つことがある。

例えば，id:int→int，abs:int→intとする．このとき，$\lambda x_{nat}.id(x_{nat})$:nat→nat$=\lambda x_{nat}.abs(x_{nat})$:nat→natである。

3　単一化と知識表現

3.1　単一化

項の制限変換の下での単一化を順序代入に基づいて次のように定義する。

定義6　任意の型付き項をt,sとする。t，sの自由変数に対する（制限変数変換の下での）代入θによって，$t\theta = s\theta$となるとき，tとsは単一化可能といい，θを単一化子という。

制限変数変換の下での変数x_σへの代入は，変数x_σをy_τと代入項t_τの型に合わせて変数変換する事に対応している。すなわち，関数変数$F_{\sigma \to \tau}$へは$F_{\sigma \to \tau} \Rightarrow H_{\sigma' \to \tau'}$なる任意のg

$\sigma' \to \tau'$が代入可能となる。特に，変数$F_{\sigma \to \sigma}$へは$t_{\tau \to \tau}, \tau \ll \sigma$，が代入できることに注意する。変数$x_\sigma$と$y_\tau$の単一化$\{x_\sigma = ? y_\tau\}$は，共通の要素を表す変数になると考えるのが自然である。即ち，$\theta^* = \{[x_\sigma := ^* z_\gamma], [y_\tau := ^* z_\gamma]\}$でUnify$\{x_\sigma = ? y_\tau\} = z_\gamma, \gamma \ll: \sigma, \gamma \ll: \tau$となる。これは両方の制限変数変換をした結果と見なすことができる。反変的な型付けではこの意味付けはできないことに注意する。反変的部分型での代入は代入項t_τを変数の型に合わせてt_σと型変換することに対応し，共変的部分型の代入と双対の関係にあるからである。さらに，反変的な型付けでは単一化後の同等性にも曖昧さが残る。このように，単一化や項の同等性＝を問題にするときには，型制約は丁度その領域型を持つと解釈すべきで，共変的部分型の定式化が適している。

例２　共変的順序の下での単一化

$F_{int \to int}$を関数変数とする．反変的継承では恒等写像$id_{nat \to nat}$は変数$F_{int \to int}$へ代入することができないが共変的順序では可能である。単一化

$$\lambda x_{nat}.F_{int \to int}(x_{nat}) = ?\ \lambda x_{nat}.id_{nat \to int}(x_{nat}) : nat \to nat$$

は成功し，単一化子$[F_{int \to int} := ^* id_{nat \to int}]$をもつ。

３．２　知識表現

型をデータ構造，型順序を知識階層構造と見なすことによって，順序ソート型理論による知識の類別的(taxonomic)記述の形式化が考えられる。基本型をソート述語[3]と見なし，各述語を型$\sigma \to o$をもつ関数と考える[10]。述語変数$P_{\sigma \to o}$をスキーマ変数，スキーマ変数を含む式をスキーマと呼ぶ。共変的部分型の下では，型$\sigma \to o$は$[\sigma]$を定義域（普遍集合）とする述語の集合と見ることができる。スキーマは単一化可能な述語の集合を特性化する高階の知識表現である。

いま，基本型としてman,human,animalを考え，man$\ll$human$\ll$animal，woman$\leqq$human$\leqq$animalとする。$\Phi_{animal \to o}$は[animal]を定義域とする述語変数となる。スキーマ$t = f(\Phi_{animal \to o})$における$\Phi_{animal \to o}$へは，$\Phi_{animal \to o} \Rightarrow^* \lambda x_{human}.\Phi_{human \to o}(x) = \Phi_{human \to o}$と制限変数変換できるから，[human]で定義された述語 $man_{human \to o}$ や$woman_{human \to o}$が代入できる。このように，制限関数意味論の下でのスキーマ表現は，論理的な知識表現[3,9,12]と親和的な記述が可能となる。特に，述語変数$P_{man \to o}$と$Q_{woman \to o}$の単一化と一般化は$R_{\perp \to o}$，$R_{animal \to o}$となる。一方，反変的な部分型の解釈では$\Phi_{animal \to o}$は少なくとも[animal]で定義された述語を表し，これと双対な知識表現になっている。特に，述語変数$P_{man \to o}$と$Q_{woman \to o}$の単一化と一般化は$R_{animal \to o}$，$R_{\perp \to o}$となる。

4 あとがき 順序ソート型付きラムダ計算における継承を新しく制限変数変換の概念を導入して定式化した。また簡約の性質について考察し，反変的部分型の継承では問題となった簡約の性質や項の同等性の記述が解消可能であること，および高階知識表現のためのスキーマ記述としても自然な枠組みであることを示した。型理論と論理系には式則是型原理のような興味ある関係が存在する。制限変数変換においても証明論的性質や可能世界モデル論的な性質を展開することが可能である。

本稿では，制限変数変換規則を変数の出現環境に依存する形で与えている。項書き替え系などとの融合 などによる定式化なども興味ある今後に残された課題である。

文献

[1]H.Ait-Kaci,A.Podelski:Towards a Meaning of LIFE, J.of Logic Programming,vol.16,pp.195-234(1993)

[2]H.Barendregt:The λ-Calculus-Its Syntax and Semantics,North Holland,Amsterdam,(1984).

[3]C.Beierle, et al An order-sorted logic for knowledge representation systems, Artificial Intelligence 55, pp.149-191 (1992).

[4]L.Cardelli:A semantics of multiple inheritance, Information and Computation76,pp.130-164,(1985).

[5]L.Cardelli,P.Wegner: On Understanding Types , Data abstraction, and Polymorphism, ACM Computing Surveys, 17, 4,pp.471-522,(1985).

[6] G.Castagna: Covariance and Contravariance: Conflict without a Cause, ACM Trans. on Programming Languages and Systems, Vol.17,No.3,May 1995,pp.431-447(1995).

[7]W.R.Cook,W.L.Hill, P.S.Canning: Inheritance Is not Subtyping, Theoretical Aspects of Object Oriented Languages,MIT Press(Ed by C.A.Gunter, J.C.Mitchell), pp.497-517,(1993)

[8]P.-L.Curien and G.Ghelli:Subtyping + Extensionality: Confluence of $\tau\eta$ top reduction in $F_{\leq}$, Proc. of Conf. Theoretical Aspects of Computer Software , Springer-Verlag, LNCS 526, pp.731-749,(1991).

[9]岩沼,原尾:様相論理に基づく知識表現と推論,人工知能学会,vol.3.No.3,pp.350-358(1988)

[10]B.Jacobs,T.Melham:Translating Dependent Type Theory into Higher Order Logic,LNCS No.664 pp.209-229,(1993)

[11] M.Kohlhase: Unification in Order-Sorted Type Theory, Proc. Int.Conf. on Logic Programming and Automated Reasoning, Springer Verlag, LNCS 624, pp.421-432,(1992)

[12]K.Meinke,J.V.Tuker(eds):Many-sorted Logic and its Applications, John Wiley &Sons,(1993).

[13]J.Meseguer:Order-Sorted Algebra Solves the Constructor-Selector, Multiple Representation, and Coercion Problem,Information and Computation, 103,pp.114-158,(1993).

[14]T.Nipkow: Higher-Order Unification,Polymorphism, and Subsorts, Proc. 2nd Int. Workshop Conditional and Typed Rewriting Systems, Springer-Verlag, LNCS 516,pp.436-447,(1991).

[15]Z.Qian: An algeraic semantics of higher-order types with subtypes,Acta Informatica,30,pp.569-607,(1993)

[16]Z.Qian,T.Nipkow:Reduction and Unification in Lambda Calculi with a General Notion of Subtype,J.of Automated Reasoning,12,pp.389-406,(1994)